



Formas de leer una ArrayList

Transcripción

[00:00] Vamos a iniciar nuestra clase número 2. Para eso, vamos a duplicar todo lo que hicimos en la clase 1. Vamos a crear una clase llamada 2, vamos a colocar aquí las variables clase 1, clase 2, clase 3, clase 4. Y ahora, vamos a ver tres formas de leer un ArrayList. La primera sería la más tradicional utilizando un for. Por ejemplo, creamos aquí un for y i que sería para la posición que queremos. Toda posición de una ArrayList comienza en la posición 0, menos listaString.

[00:49] Y como en la clase anterior ya vimos, tenemos el método size para ver el tamaño de nuestra lista. Y vamos a dar un i++. Una vez hecho esto, vamos a dar un System.out.println y vamos a imprimir el valor de nuestra lista.

ListaString.get para obtener el valor de la posición de nuestra lista tal. Por ejemplo lista string, queremos la posición i, que comenzaría con 0. Vamos a ejecutar. Imprimió clase 1, 2, 3 y 4.

[01:40] Luego, tenemos otra forma de leer una lista. Podemos utilizar un for each, que en realidad es un for, solamente que la nomenclatura de él es un poco diferente. ¿Cómo utilizo estos dos puntos? Entonces para eso aquí al inicio tenemos que declarar nuestra variable. ¿Cuál es nuestra variable que queremos? Vamos a colocar clase. Y queremos que recorra listaString.

[02:20] Eso quiere decir que todos esos valores van a ser colocados en ese string clase y va a hacer un loop. Entonces aquí vamos a hacer lo siguiente. Vamos a imprimir System.out.println y digitamos clase. Una vez hecho esto, vamos a ejecutar nuevamente. Imprimió clase 1, 2, 3 y 4. ¿Qué hizo aquí? Recorrió

nuestra `listaString`, colocó cada valor, comenzó a colocar la clase 1 y comenzó a imprimir en ella.

[03:09] Ahora a partir de Java 8 tenemos un recurso llamado `lambda`, que también nos va a permitir leer una lista de una forma diferente. Por ejemplo, pongo aquí `listaString.forEach`, colocamos. ¿Cuál sería nuestra variable? Vamos a colocar `clase`. Abrimos, cerramos llaves, que sería, todo esto de aquí es un padrón de `lambda`, e imprimimos: `System.out.println`. ¿Y qué colocamos? El valor, que sería `clase`.

[03:54] Ahora vamos a ejecutar nuevamente y tenemos aquí `clase 1`, `clase 2`, `clase 3` y `clase 4`. Entonces estamos viendo aquí tres formas de leer una lista utilizando un `for` tradicional, utilizando un `for` tipo `forEach` y utilizando un `forEach` recurso de `lambdas`, que es solamente a partir de Java 8. Esto sería todo nuestra `clase número 2`.